第 32 卷 第 6 期 吉首大学学报(自然科学版) Vol. 32 No. 6
2011 年 11 月 Journal of Jishou University (Natural Science Edition) Nov. 2011

文章编号:1007-2985(2011)06-0055-04

基于聚类技术的 XML 文件代表性结构获取*

卓月明
(吉首大学软件服务外包学院,湖南 张家界 427000)

摘 要:XML 文件可以利用树状结构来表示,于是把如何将 XML 文件做聚类看成如何对树状结构的数据作聚类.使用 SOM 聚类工具搭配上 Jaccard 的距离测量公式来对 XML 文件做聚类,然后在每个 cluster 中利用 GST(Graph Search Technique)算法从这些 XML 文件当中找出他们的最大序列,最后将这些最大序列融合起来成为共同的结构.

关键词:XML 文件;树状结构;聚类;序列挖掘;相同结构
中图分类号:TP311 **文献标志码**:A

XML 技术在当前的互联网络和 IT 环境中扮演着越来越中重要的角色,它实际上已经成为数据交换的标准、SOA 架构的基石.一般来说,整合不同的 XML 文件通常通过 DTD 或 XSD,但不能保证能够同时得到 XML 文件和其结构信息.有许多关于 XML 文件的聚类的研究,其中有些是根据计算 XML 文件之间的编辑距离来计算他们的相似度[1],而也有一些是基于向量空间模型来处理[2].笔者利用了广受欢迎的 SOM(Self-Organizing Map)与 Jaccard 系数作为衡量的距离,对 XML 文件根据其结构进行聚类操作,并进一步找出它们的共同结构.在每个 cluster 中,笔者利用 GST(Graph Search Technique)[3]从 XML 文件中提取最大频繁序列,然后合并它们产生的共同结构.

1 相关工作

1.1 XML 的树状结构

XML 文件如下:

```
〈Book〉
  〈Author〉
    〈Name〉Mark〈/Name〉
    〈Phone〉01234567〈/Phone〉
    〈Email〉mark@163.com〈/Email〉
  〈Author〉
  〈Publication〉
    〈year〉2011〈Year〉
    〈Number〉300〈/Number〉
  〈/Publication〉
〈/Book〉
```

用树状的结构来表示 XML 文件如图 1.不考虑 XML 里面出现的 IDREFS 和 Hyperlinks,并且,对于文件内的元素和属性都以其卷标作为树状结构内节点的名称.由于目标在于根据相似的结构来找出共同的结构,所以笔者只考虑 XML 文件的结构,而不考虑其内容.

1.2 树的包含关系

根据文献[4],XML 文件的树状关系可以被分成 3 种不同的概念:subtree inclusion,tree embedding 和 tree subsumption.

(1) Subtree Inclusion. Subtree inclusion 的关系是最严格的一种定义.设有 2 棵树 X 和 Y,根据 subtree inclusion 的定

* 收稿日期:2011-06-21
作者简介:卓月明(1970-),男,湖南慈利人,吉首大学软件服务外包学院副教授,硕士,主要从事数据库和智能计算研究.

义,Y 可以被包含在 X 里面,其条件为在 X 里面必须有 1 棵子树是跟 Y 长得一模一样. 在图 2 中,T'和 T1 构成 subtree inclusion 关系.

(2) Tree Embedding. 同样假设有 2 棵树 X 和 Y,Y 根据 tree embedding 的定义被包含在 X 内,必须满足:1)在 Y 里面的所有 node 必须被保存在 X 里面;2)在 Y 里面的所有祖孙关系必须被保存在 X 里面;3)在 Y 里面没有出现的关系,同样也不能出现在 X 内. 在图 2 中,可以说 T'根据 tree embedding 被包含在 T1,T2 里面.

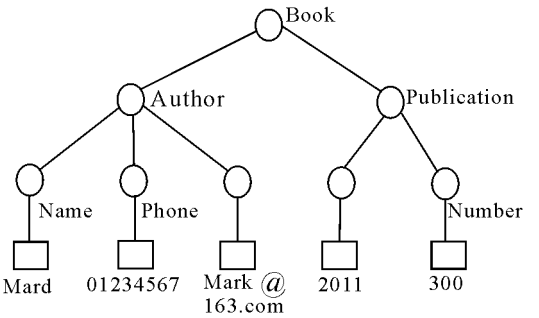


图 1 XML 文件和其相对应的树状结构

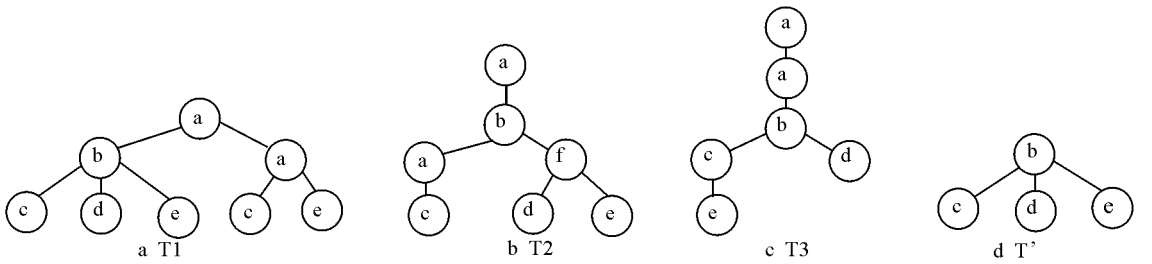


图 2 各种不同的 XML 树状结构

(3) Tree Subsumption. 同样假设有 2 棵树 X 和 Y,Y 根据 tree subsumption 定义被包含在 X 内,必须满足:1)在 Y 里面的所有 node 必须被保存在 X 里面;2) 在 Y 里面的所有祖孙关系必须被保存在 X 里面. 在图 2 中,根据 tree subsumption 定义,T'可以被包含在 T1 和 T2 内. 笔者希望找出来的共同结构是能够尽可能找出最大的,所以笔者采用的是 tree subsumption 的定义.

1.3 序列模式挖掘应用于树状结构

序列模式挖掘是由 Rakesh 及 Ramakrishnan[5] 在 1995 所提出来的,给定一个事务数据库,可以找出满足最小支持度的序列模式.

1.4 图形搜寻算法

图形搜寻算法可以不用通过($k-1$)-sequences 的序列去找出 k-sequences($k>=3$)[6],所以效率是比传统 Apriori-like 的算法要好. 首先,图形搜寻算法先利用 2-sequences 产生 item relation graph(IRG),再利用搜寻的技巧在 IRG 里面找出所有的 k-sequences ($k>=3$).

2 系统架构

系统主要分成 2 部分:聚类部分和挖掘部分,其结构图如图 3 所示.

2.1 聚类部分

首先将每份 XML 文件以 label-pair 的形式转换成矩阵,然后再利用 SOM 的聚类方法将其作聚类.

(1) 矩阵转换. 一份 XML 文件可以被看成树状结构,对于 1 个树状结构来说,若节点跟节点之间有 1 条路径,则可以取得节点跟节点之间的关系以 $x*y$ 的方式来呈现;x 为 y 的祖先,y 为 x 的孙子. 举个例子来说,在图 2 中的 T1 可以用 T1={a*b,a*c,a*d,a*e,a*a,b*c,b*d,b*e}的形式来取代. 为了决定 XML 文件之间的相似度,label-pairs 在这边扮演着一个重要的角色,不只考虑了节点,也考虑了节点跟节点之间的连接关系. 将所有的 XML 文件转换完之后,会将它们融合成一个矩阵 DM(n,m),n 为 XML 文件的总数量,m 为所有的 label-paris. 其中 label-pair 可以被视为是每份 XML 文件的特征,后面将会用这些特征来对这些 XML 文件做聚类.

(2) 聚类方法. 有了矩阵之后,普可以用 SOM 的聚类方法来做聚类. SOM 是由 Kohonen 于 1980 提出的,是一种由

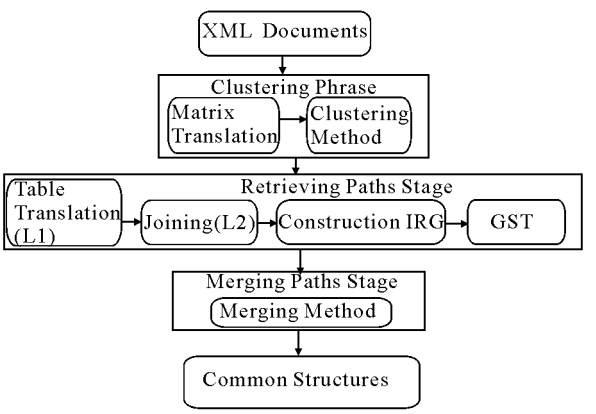


图 3 系统结构

任意点组织成 1 个拓扑图的非监督式的聚类方法. 笔者采用 Jaccard coefficient 的距离测公式. Jaccard coefficient 的值是介于 0～1 之间,0 表示 2 个对象为完全不相同,1 则表示 2 个对象完全相同. 举个例子来说,5 个对象以及各自的特征如表 1,并且他们的 Jaccard 相似被计算在如表 2.

表 1　5 个对象及其特征

Set	Feature				
A	1	1	0	1	1
B	1	0	1	0	0
C	0	1	0	1	1
D	1	0	1	0	1
E	1	0	1	1	1

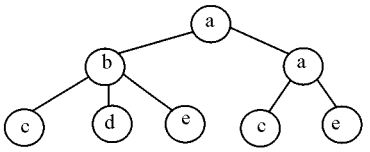
表 2　Jaccard 相似

	A	B	C	D	E
A	1.00	0.20	0.75	0.40	0.60
B	0.20	1.00	0.00	0.67	0.50
C	0.75	0.00	1	0.20	0.40
D	0.40	0.67	0.20	1.00	0.75
E	0.60	0.50	0.40	0.75	1.00

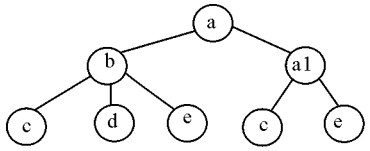
2.2 挖掘部分

2.2.1 取得序列　为了去取的最长的共同路径,则有如下 4 个步骤:

(1) 将 XML 文件转换成 XML Table 并产生 Large 1-sequences. 如在图 2 中,T1 有 1 条路径 a-a-c,所以这 2 个标签 a 将会把它重新命名为 a1 并且将其存入 MapTable 里,如图 4 所示,最后将 T1 转换成表 3.



a The same lable appears in one path



a Rename

a	a1
b	
c	
d	
e	

c Maptable

图 4　MapTable

表 3　XML_table

Doc_name	Path_No	Level_No	Node_Name	Doc_name	Path_No	Level_No	Node_Name
T1	1	1	a	T1	3	3	e
T1	1	2	b	T1	4	1	a
T1	1	3	c	T1	4	2	a1
T1	2	1	a	T1	4	3	c
T1	2	2	b	T1	5	1	a
T1	2	3	d	T1	5	2	a1
T1	3	1	a	T1	5	3	e
T1	3	2	b				

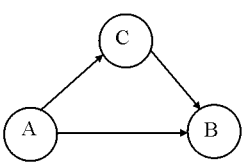
再将同 1 个 cluster 内所有的 XML 文件转换成 XML_table 之后,就有了 C1. 在计算是否符合最小支持度的时候,若遇到有重新命名过的标签,必须通过 MapTable 将它当成是尚未重新命名之前的标签来计算,最后能够得到 L1 进而去得到 C2.

(2) Joining L1 本身去产生 Large 2-sequences. 用在 L1 内的 Doc_Name,Path_No,and Level_No 去对 L1 做两两组合去产生 L2. C2 和 L2 的格式如图 5. 每个 2-sequence 包含 2 个按照顺序的卷标,第 1 个卷标其 Level_No 比第 2 个标签小,并且他们的 Level_No 将会各自存在 Level_Start 和 Level_End 内. 在过滤完所有不满足最小支持度的 2-sequences 之后,就可以得到 L2.

Sequence	Dod_Name	Path_No	Level_Start	Level_End

图 5　C2 和 L2 的格式

(3) 建构 IRG. 在有 L2 之后,可以建构出 IRG,其中的节点及为在 L2 里面 2-sequence 的标签名称,并且在边上也会纪录卷标之间的顺序. 对于 1 个 2-sequence 〈(A)(B)〉,会画一条实线箭头从 A 到 B 并且每个边包含可以在 L2 得到的信息 Doc_Name,Path_NO,Level_Start,和 Level_End 如同图 6 所示.



Doc_Name	Path_No	Level_No	Node_Name

图 6　〈(A)(B)〉,〈(A)(C)〉和 〈(C)(B)〉的 IRG 以及它的边信息

(4) 使用图形搜寻算法去产生最大序列. 构建一个 IRG 后,采用在 DFS(深优先搜)的方法来查找大型序列,并进一步寻求最大频繁序列处理过程与原 GST 算法大部分相同,但在使用 GST 时,必须检查在 IRG 上面的 2 个边是否真在原始文

件上面是连续的找出最大频繁序列的算法如下：

```
S:Set of large sequences
for(k=n;k>1;k--)
for each k-sequence Sk
delete the subsequence of Sk from S
return S
```

2.2.2 合并路径　在这个阶段，合并在 1 个 cluster 内的最大频繁序列并产生其共同的结构. 在合并路径时，若新加入 1 个节点到共同结构内，必须检查是否符合 tree subsumption 的定义；如果没有符合 tree subsumption 的定义，会将产生的共同结构树拆成 2 个部份，并分别以递归的方法继续

3 结语

笔者提出了一种有效的方法，从多个 XML 文件去提取它们的共同结构. 利用 SOM 去对 XML 文件作聚类然后，利用改良的循序挖掘技术(图形搜寻技术)去找出最大频繁序列，最后合并这些最大频繁序列并产生共同结构.

参考文献：

[1] WEI Jin-mao, WANG Shu-qin, WANG Jing, et al. Fast Kernel for Calculating Structural Information Similarities [C]//Proc. the 3rd IEEE International Conference on Intelligent Systems. Varna, Bulgaria, 2006: 59-64.

[2] YANG Jian-wu, WILLIAM K CHEUNG, CHEN Xiao-ou. Learning the Kernel Matrix for Xml Document Clustering [C]//Proc. the IEEE International Conference on e-Technology, e-Commerce and e-Service. Hong Kong, China, 2005: 353-358.

[3] HUANG Yin-fu, LIN Shao-yuan. Mining Sequential Patterns Using Graph Search Techniques [C]//Proc. the 27th Annual International Conference on Computer Software and Applications. Washington, DC, USA, 2003: 4-9.

[4] ALEXANDRE TERMIER, MARIE-CHRISTINE ROUSSET, MICHèL SEBAG. TreeFinder: A First Step Towards Xml Data Mining [C]//Proc. the IEEE International Conference on Data Mining. Maebashi City, Japan, 2002: 450-457.

[5] AGRAWAL R, SRIKANT R. Mining Sequential Patterns [C]//Proc. the 11th IEEE International Conference on Data Engineering. Taipei, Taiwan, 1995: 3-14.

[6] HUANG Yin-fu, LIN Shao-yuan. Mining Sequential Patterns Using Graph Search Techniques [C]//Proc. the 27th Annual International Conference on Computer Software and Applications. Washington, DC, USA, 2003: 4-9.

Representative Structures from XML Documents Based on Clustering Techniques

ZHUO Yue-ming

(Software & Outsourcing Institute, Jishou University, Zhangjiajie 427000, Hunan China)

Abstract: Since an XML document can be represented as a tree structure, the problem how to cluster a collection of XML documents can be considered as how to cluster a collection of tree-structured documents. The author used SOM (Self-Organizing Map) with the Jaccard coefficient to cluster XML documents. Then, an efficient sequential mining method called GST was applied to find maximum frequent sequences. Finally, the author merged the maximum frequent sequences to produce the common structures in a cluster.

Key words: XML document; tree-structured; clustering; sequential pattern mining; common structure

(责任编辑　陈炳权)